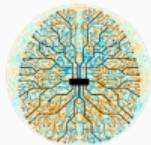


# Machine Learning for Radiometer Calibration

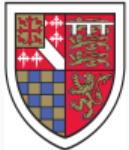
Samuel Alan Kossoff Leeney

2nd Year PhD Candidate

With: Harry Bevins, Eloy de Lera Acedo, Will Handley, Rohan Patel, Kaan Artuc, Jiacong Zhu



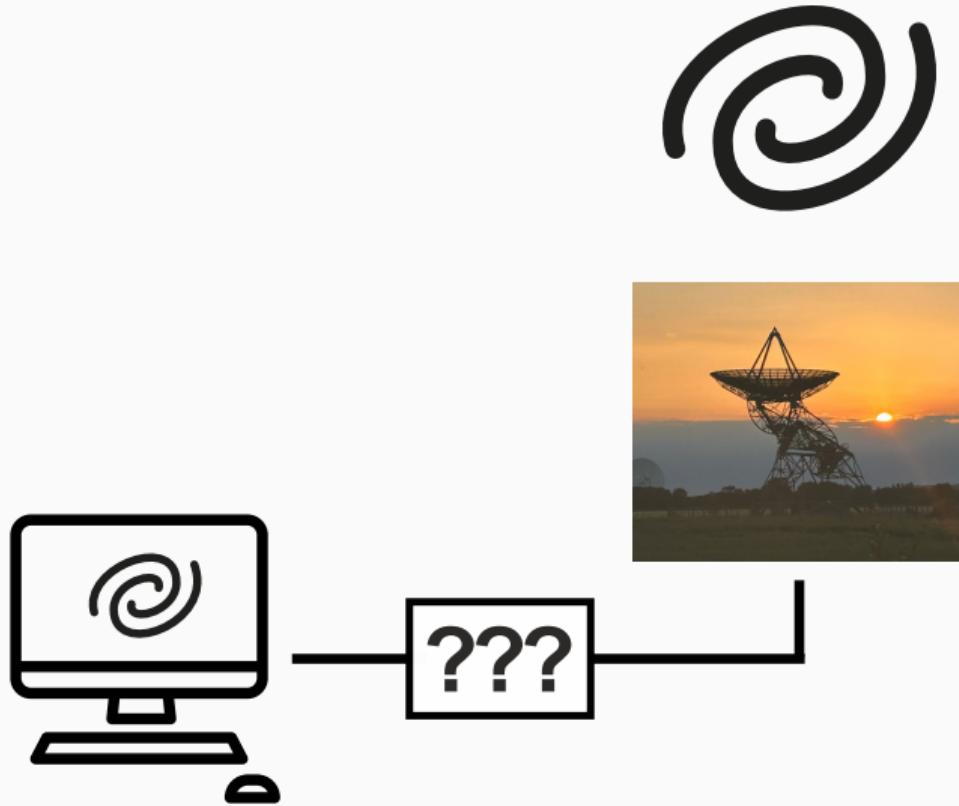
UNIVERSITY OF  
CAMBRIDGE



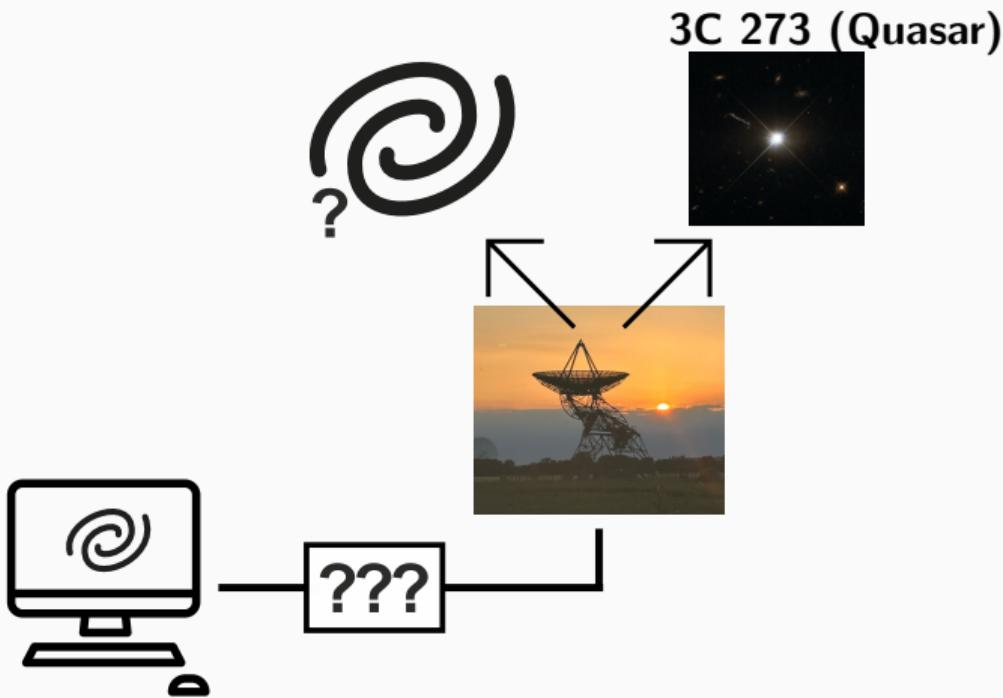
## **Radiometer calibration overview**

---

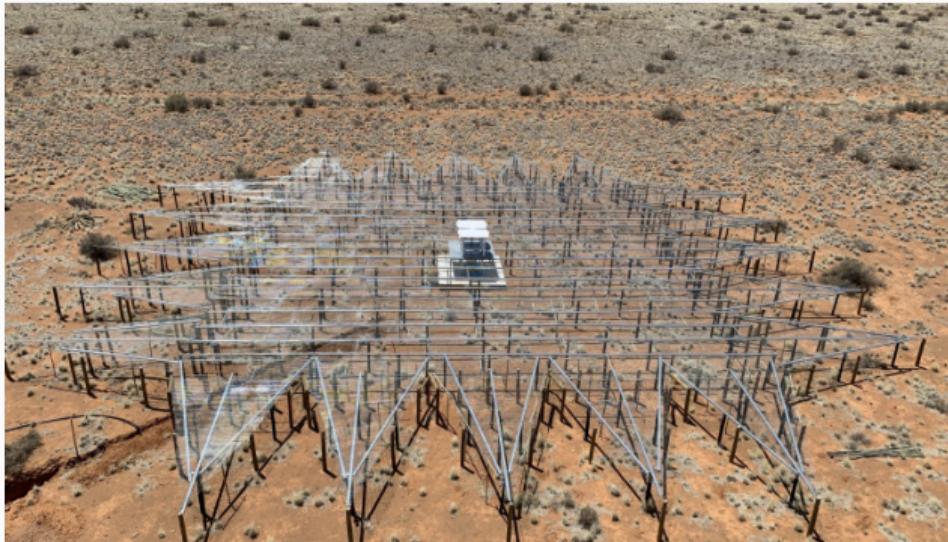
# What is calibration?



## How to calibrate?



## Why is calibration in Global 21cm Cosmology difficult?



We measure *sky averaged* signal.

Antenna LNA impedance mismatch

Very faint signal.

## How to calibrate (in a bit more detail...)?

**Objective:** Map input temperature to output power.

### Key Factors:

- LNA introduces time-dependent gain,  $g(t)$ .
- Impedance mismatch adds noise ( $T_{\text{rec}}$ ) to the system.

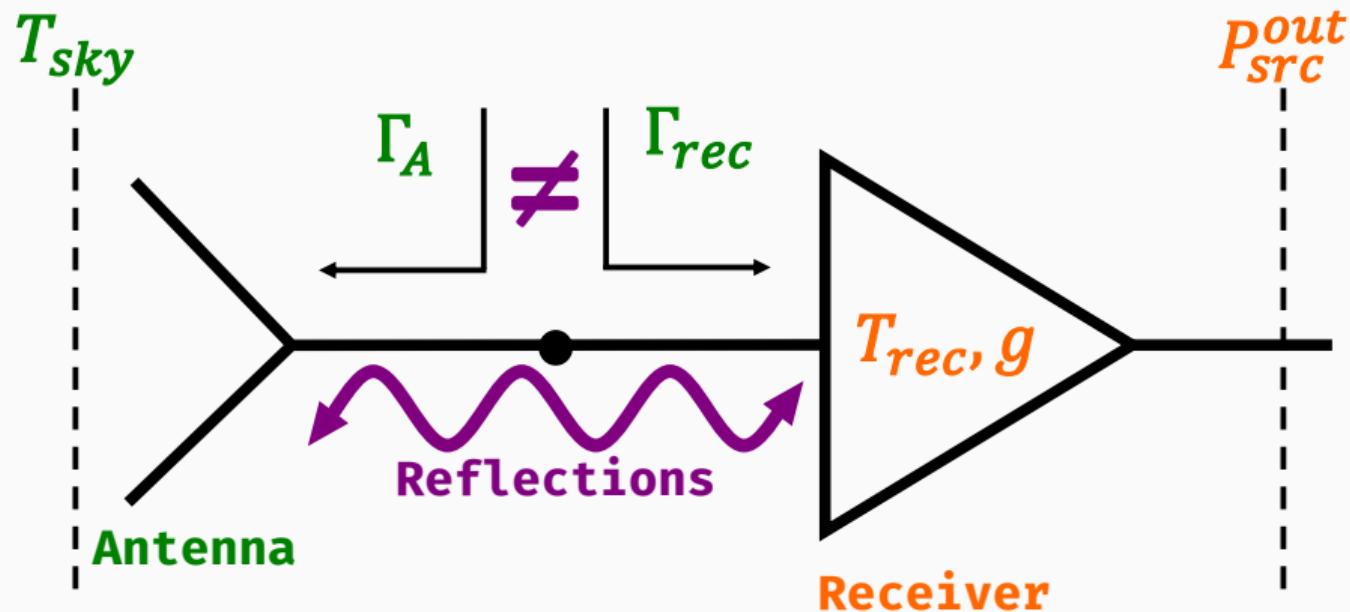
### Link Output Power to Input Temperature:

$$P_{\text{out}}^{\text{src}} = gM \times (T_{\text{in}}^{\text{src}} + T_{\text{rec}}) \quad (1)$$

$$M = \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{(1 - |\Gamma_{\text{cal}}|^2)} | \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{1 - \Gamma_{\text{cal}} \Gamma_{\text{rec}}} |^2$$

*Note: All parameters above are frequency-dependent, but the notation has been simplified here and thereafter for convenience.*

## Receiver/source mismatch



**Figure 1:** Schematic of the receiver system showing the signal path from antenna to digitizer

# Dealing with reflections...

$$P_{\text{out}}^{\text{src}} = gM (T_{\text{in}}^{\text{src}} + T_{\text{rec}}) \quad (2)$$

## Noise Parameter Equation:

$$P_{\text{out}}^{\text{src}} = gM \left( T_{\text{in}}^{\text{src}} + T_{\text{min}} + T_0 \frac{4R_N}{Z_0} \frac{|\Gamma_{\text{src}} - \Gamma_{\text{opt}}|^2}{(1 - |\Gamma_{\text{src}}|^2)(1 + |\Gamma_{\text{opt}}|^2)} \right) \quad (3)$$

## Noise Wave Equation:

$$\begin{aligned} P_{\text{out}}^{\text{src}} = & g \left[ T_0 + T_{\text{unc}} |\Gamma_s|^2 \left| \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{1 - \Gamma_s \Gamma_{\text{rec}}} \right|^2 \right. \\ & + T_s (1 - |\Gamma_s|^2) \left| \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{1 - \Gamma_s \Gamma_{\text{rec}}} \right|^2 + T_{\cos} \Re \left( \Gamma_s \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{1 - \Gamma_s \Gamma_{\text{rec}}} \right) \\ & \left. + T_{\sin} \Im \left( \Gamma_s \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{1 - \Gamma_s \Gamma_{\text{rec}}} \right) \right] \quad (4) \end{aligned}$$

- $g$  - Gain
- $T_{\text{min}}$  - Minimum Noise Temperature
- $R_N$  - Noise Resistance
- $\Gamma_{\text{opt}}$  - Optimum Reflection Coefficient
- $\Gamma_s$  - Source Reflection Coefficient
- $\Gamma_{\text{rec}}$  - Receiver Reflection Coefficient
- $T_s$  - Source Temperature
- $P_{\text{out}}^{\text{src}}$  - Power out
- $T_{\text{unc}}, \cos, \sin$  - Noise wave parameters
- $T_0$  reference temperature

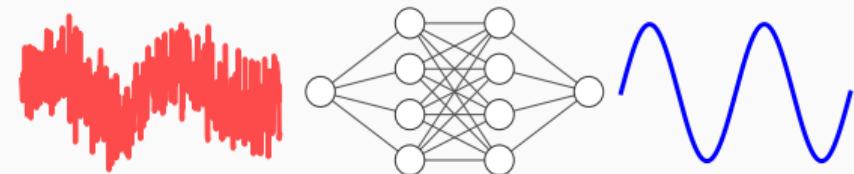
# **Radiometer calibration with machine learning**

---

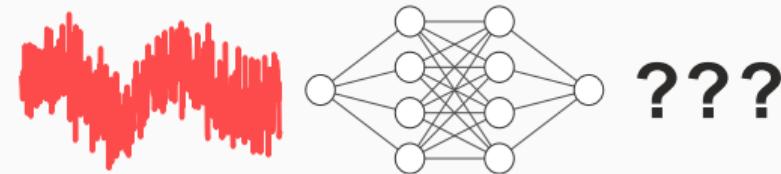
## How/why?

- Can predict noise wave parameters using machine learning.
- Could apply method to noise parameters.
- Predict **directly** on noise parameters on frequency by frequency basis.
- Maleable to environmental changes.

Train



Predict



# Machine learning calibration steps

## 1. Define the Loss Function

Regress over measured power and predicted power.

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^n (\mathcal{P}_{\text{measured},i} - \mathcal{P}_{\text{pred},i})^2 \quad (5)$$

## 2. Write Down the Equation for $\mathcal{P}_{\text{pred}}$

Using the noise wave formalism, relate  $\mathcal{P}_{\text{pred}}$  to  $T_{\text{src}}$ .

$$\begin{aligned} \mathcal{P}_{\text{pred}} &= \mathbf{g} \cdot M(T_{\text{in}}^{\text{src}} \\ &+ T_{\text{min}} + T_0 \frac{4R_N}{Z_0} \frac{|\Gamma_{\text{src}} - \Gamma_{\text{opt}}|^2}{(1 - |\Gamma_{\text{src}}|^2)(1 + |\Gamma_{\text{opt}}|^2)} \Big) \end{aligned} \quad (6)$$

## 3. Minimise the Loss Function

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta) \quad (7)$$

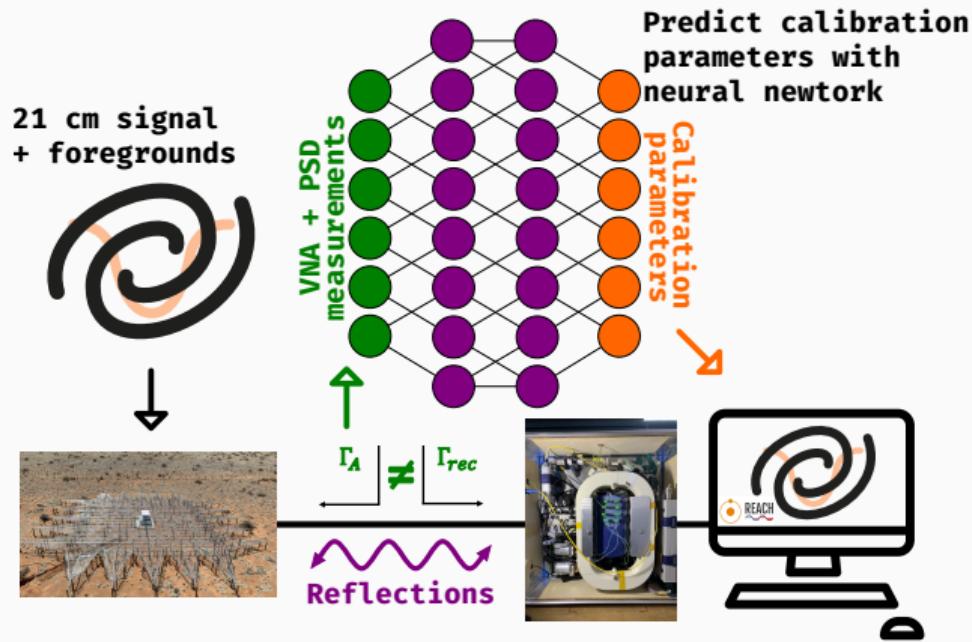
parameter vector  $\theta$  includes all tunable parameters in the model:

$$\theta = \{\mathbf{g}, T_{\text{min}}, R_N, \Gamma_{\text{opt}}^\phi, |\Gamma_{\text{opt}}|\} \quad (8)$$

## 4. Rearrange and Predict ( $T_{\text{src}}$ ) using $\theta^*$

$$\begin{aligned} T_{\text{src}} &= \frac{\mathcal{P}_{\text{pred}}}{\mathbf{g}^* \cdot M} \\ &- \left( T_{\text{min}}^* + T_0 \frac{4R_N^*}{Z_0} \frac{|\Gamma_{\text{src}} - \Gamma_{\text{opt}}^*|^2}{(1 - |\Gamma_{\text{src}}|^2)(1 + |\Gamma_{\text{opt}}^*|^2)} \right) \end{aligned} \quad (9)$$

## ML calibration overview

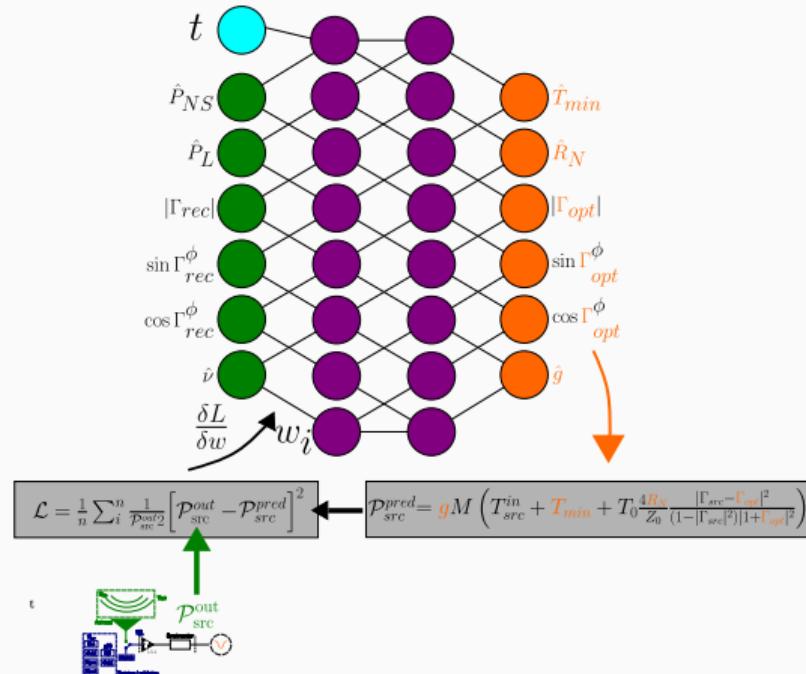


**Figure 2:** High-level overview of the machine learning-based calibration framework

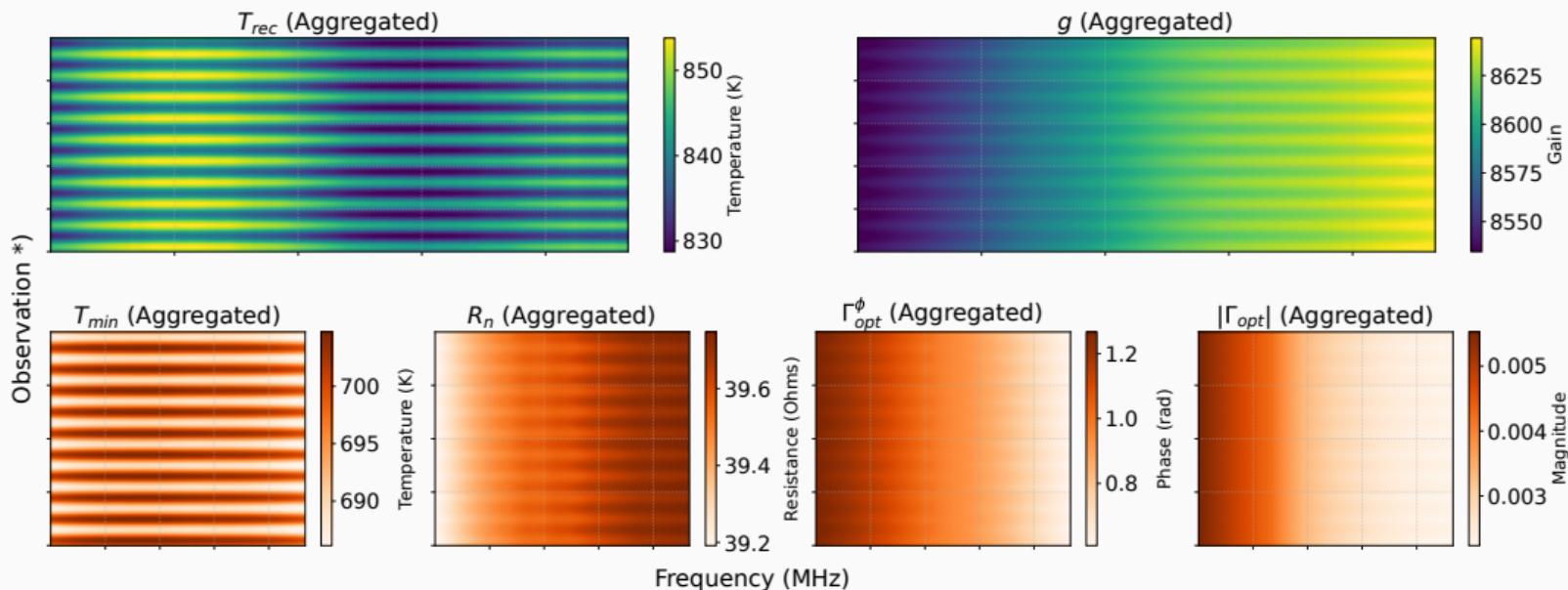
## **Learning non-linear time-dependent system drift**

---

# Neural Network Time Evolution

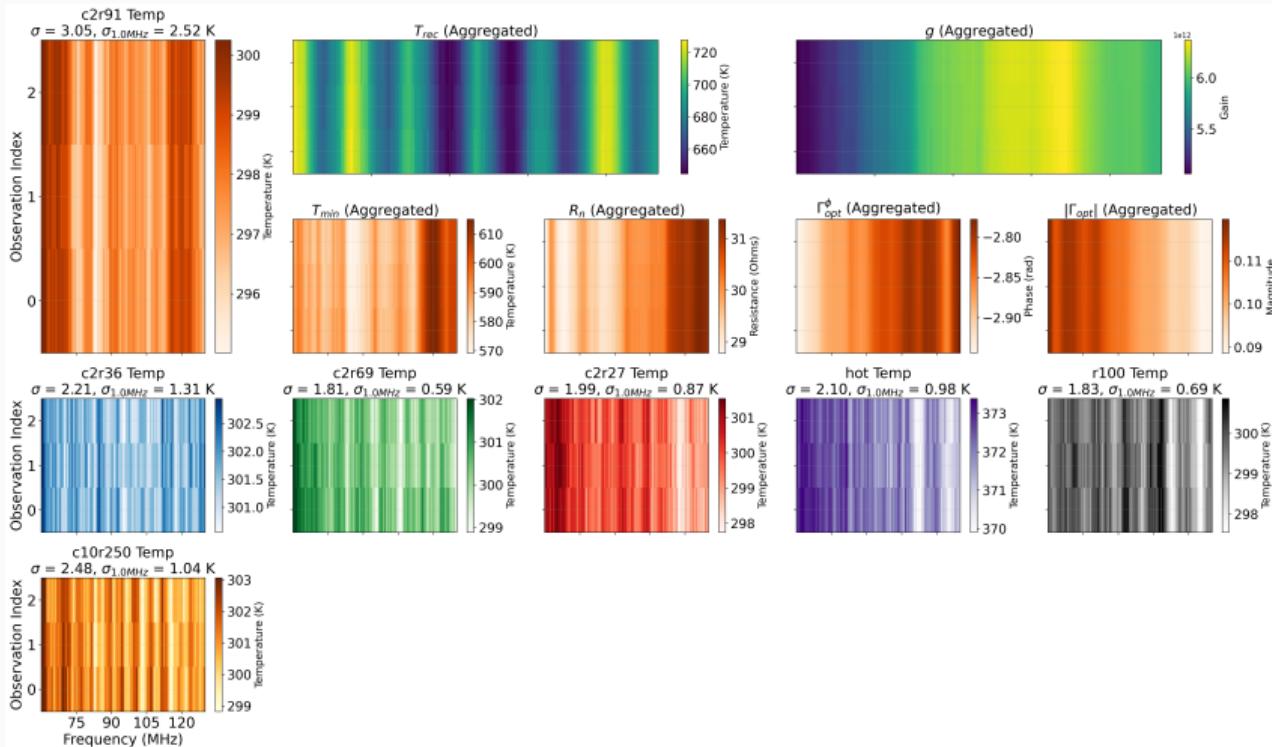


## Inject system drift

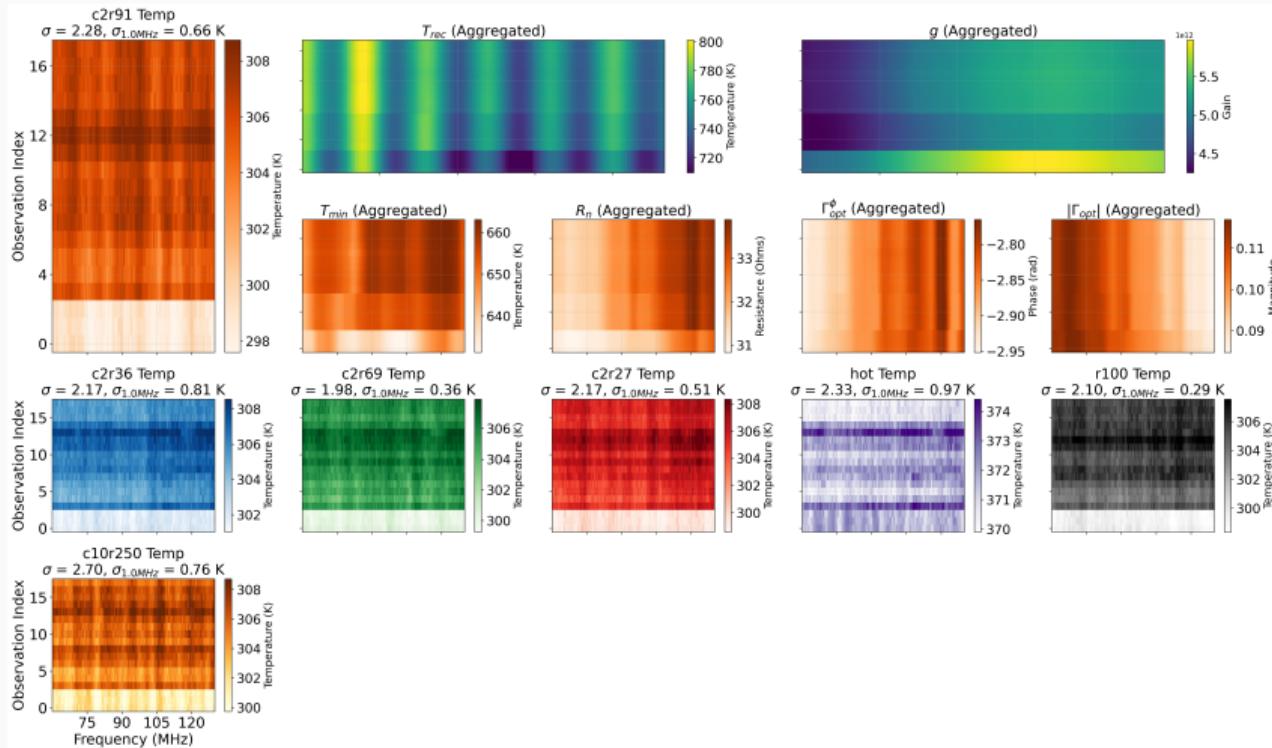


Inject a time varying sinusoid into  $T_{min}$  and predict the noise parameters → the network recovers this 'system drift'

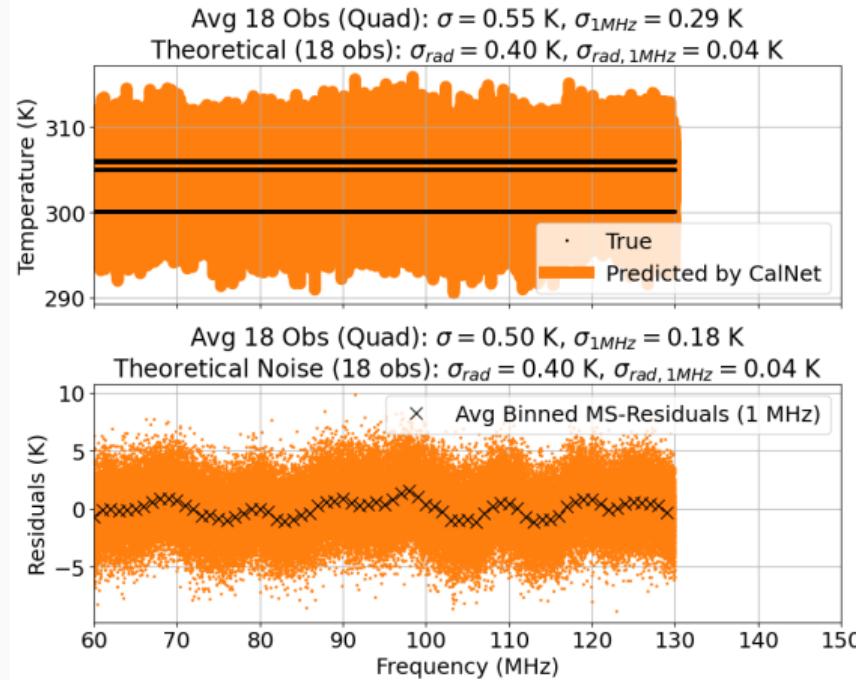
# Single night, many observations



# Many nights, many observations



# Calibration Performance

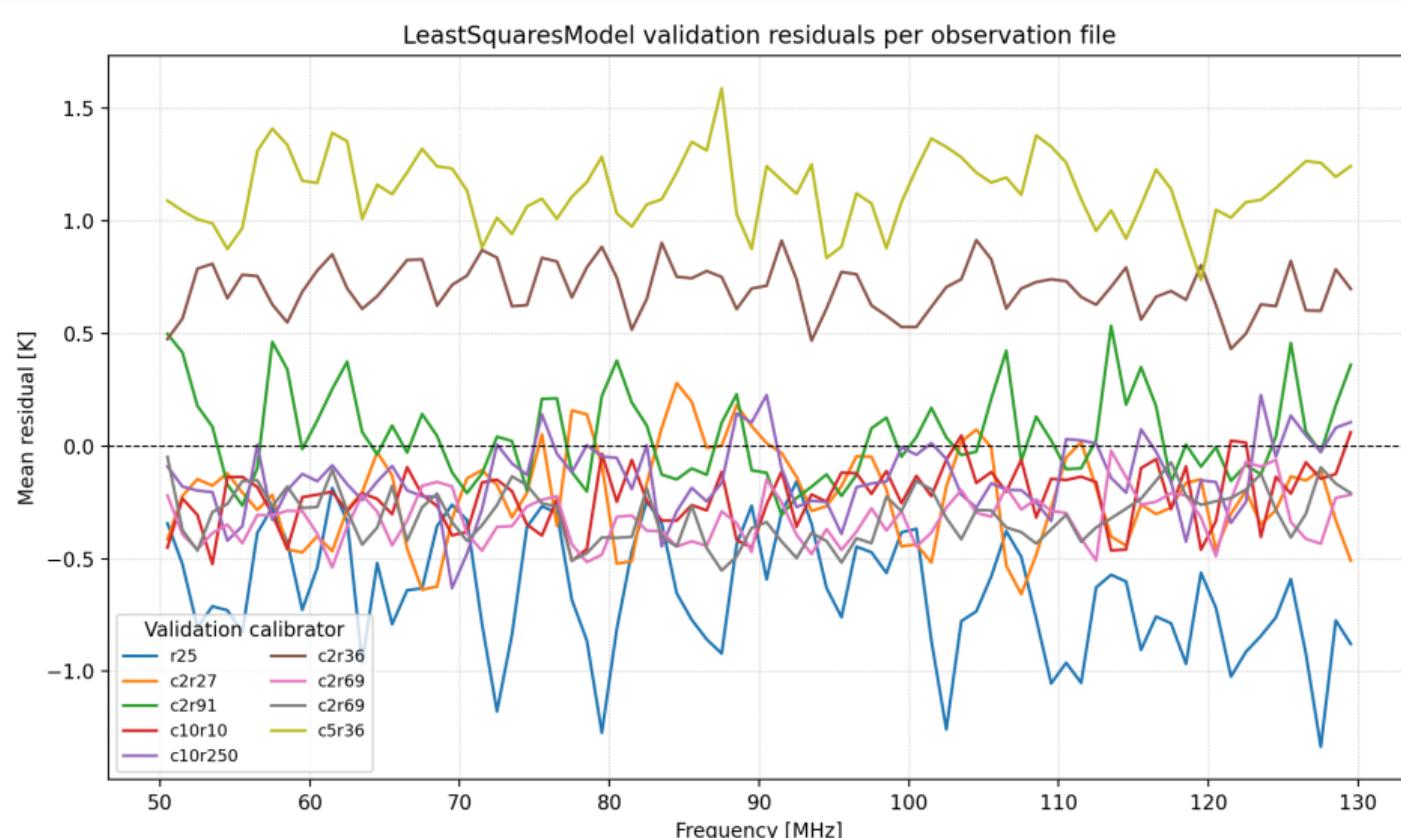


- Significant improvement when combining many nights
- Calibration down to 0.18K
- Getting close to theoretical noise

## **Neural corrections**

---

# Neural LSQ residual diagnostics



## Neural pre-corrected LSQ pipeline

1. Neural network predicts additive corrections:  $\Delta T = \text{NN}(f, \Gamma)$ .
2. Apply the correction to the measured spectrum:  $T_{\text{corrected}} = T_{\text{measured}} + \Delta T$ .
3. Run a least-squares solve inside the loss:  $\theta = \arg \min_{\theta} \|X\theta - T_{\text{corrected}}\|^2$ .
4. Predict the calibrated temperature:  $T_{\text{pred}} = X\theta$ .
5. Define the loss:  $\mathcal{L} = \|T_{\text{pred}} - T_{\text{corrected}}\|^2$ .

The LSQ solver is executed *inside* the training loop, so JAX backpropagates through the closed-form solve.

## Neural pre-corrected LSQ mathematics

The LSQ design matrix for frequency  $f$  and calibrator  $c$  is

$$X_c = \begin{bmatrix} x_u & x_c & x_s & x_{NS} & x_L \end{bmatrix},$$

where the noise-wave formulation provides  $X$  from the calibrator and receiver  $S_{11}$ .

For each frequency we solve

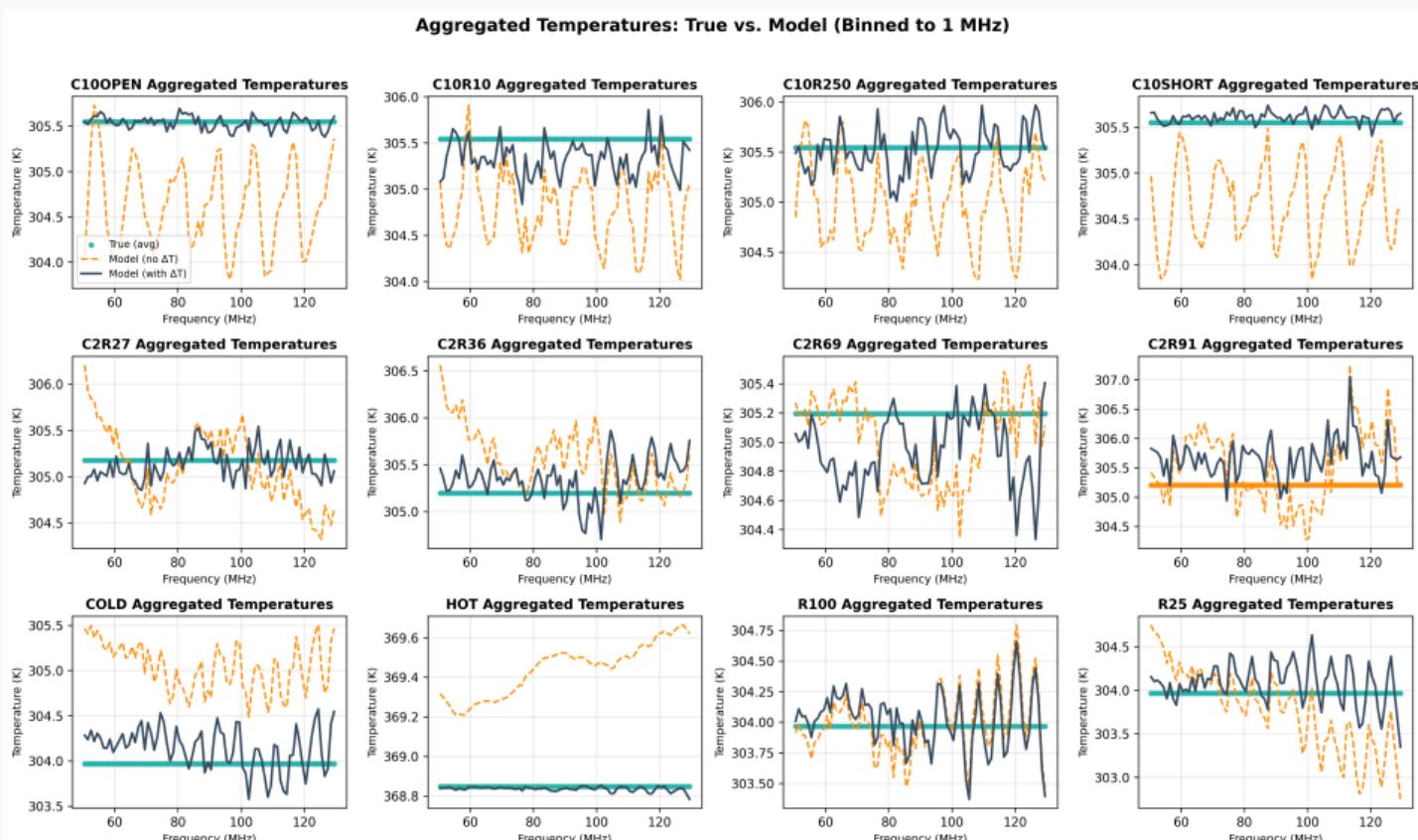
$$\theta(f) = \arg \min_{\theta} \|X(f)\theta - (T_{\text{measured}} + \text{NN}(f, \Gamma; w))\|^2,$$

and the training loss becomes

$$\mathcal{L}(w) = \sum_{f,c} [T_{\text{measured}}(c) - (X_c(f) \cdot \theta(f) - \text{NN}(f, \Gamma_c; w))]^2.$$

Optimising  $w$  therefore yields the neural correction weights that minimise the residual between corrected measurements and LSQ predictions.

# Aggregated calibrator temperatures



## **Neural anomaly detection**

---

## Bayesian anomaly detection problem

- Each data point carries an anomaly mask  $\varepsilon_i \in \{0, 1\}$  with Bernoulli prior  $P(\varepsilon_i) = p_i^{\varepsilon_i} (1 - p_i)^{1 - \varepsilon_i}$ .
- Likelihood blends the normal model  $\mathcal{L}_i(\theta)$  and a uniform anomaly floor  $\frac{1}{\Delta}$ :

$$P(\mathcal{D}, \vec{\varepsilon} | \theta) = \prod_{i=1}^N [\mathcal{L}_i(\theta)(1 - p_i)]^{1 - \varepsilon_i} \left[ \frac{p_i}{\Delta} \right]^{\varepsilon_i}.$$

- Dominant mask approximation yields the rule  $\log \mathcal{L}_i + \log \Delta \leq \text{logit}(p_i)$  for deciding whether a point is anomalous.

# Hard vs. soft mixture models

## Hard mixture:

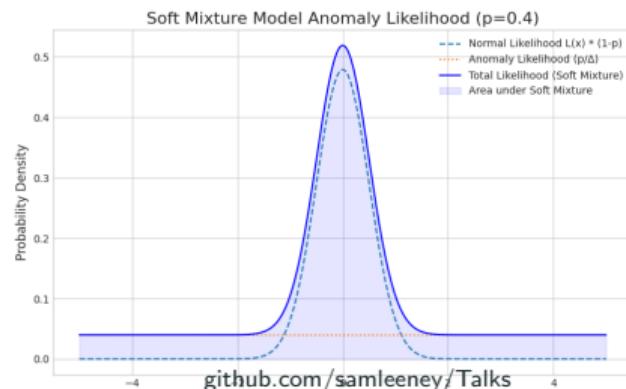
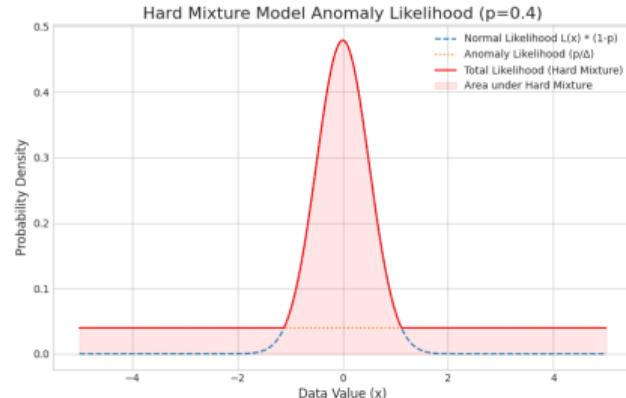
$$\log P(\mathcal{D}|\theta) = \sum_i \max \{ \log \mathcal{L}_i + \log(1 - p_i), \log p_i - \log \Delta \}$$

- Fast but clips between hypotheses.
- Sensitive when  $p_i$  varies with environment.

## Soft mixture:

$$\log P(\mathcal{D}|\theta) = \sum_i \log \left[ (1 - p_i) \mathcal{L}_i + \frac{p_i}{\Delta} \right]$$

- Smoothly marginalises corruption states.
- Posterior anomaly probability drops below 0.5 when  $(1 - p_i) \mathcal{L}_i < \frac{p_i}{\Delta}$ .



## Neural anomaly architecture and loss

- Neural network outputs both regression prediction  $\hat{y} = f_\theta(x)$  and anomaly prior  $\text{logit}(p_i) = g_\theta(x)$ .
- Hard-loss training maximises  $\sum \max(\log \mathcal{L}_i + \log(1 - p_i), \log p_i - \log \Delta)$ , producing binary masks similar to earlier Bayesian RFI mitigation methods.
- Soft-loss training uses a `logsumexp` implementation for numerical stability:

$$\mathcal{L}(\theta) = \sum_{i=1}^N \text{logsumexp} (\log \mathcal{L}_i + \log(1 - p_i), \log p_i - \log \Delta).$$

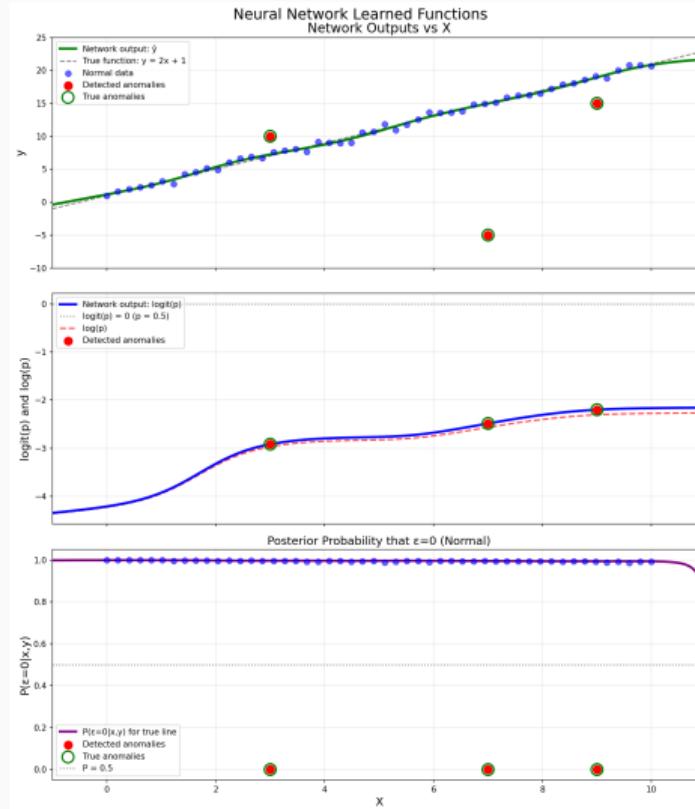
- Posterior anomaly probability follows

$$P(\varepsilon_i = 0 | x_i, y_i, \theta) = \frac{(1 - p_i)\mathcal{L}_i(\theta)}{(1 - p_i)\mathcal{L}_i(\theta) + \frac{p_i}{\Delta}},$$

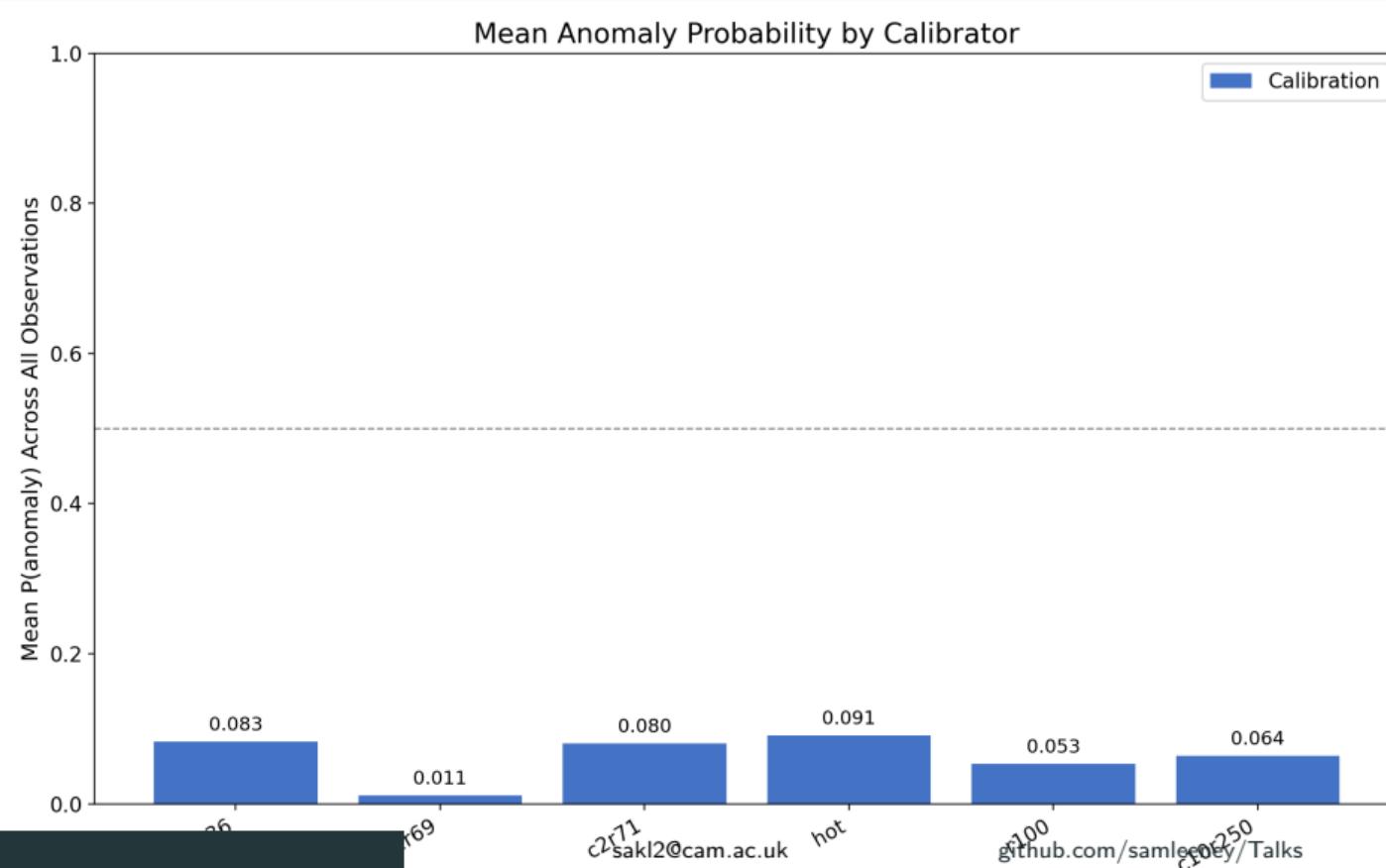
giving interpretable anomaly maps for the receiver chain.

# Toy anomaly detection results

- Synthetic linear regression with Gaussian noise plus injected RFI-like anomalies.
- Network jointly fits  $\hat{y}$  and spatial  $\text{logit}(p)$ , achieving perfect 3/3 anomaly recall with zero false positives.
- Posterior  $P(\varepsilon = 0|x, y)$  cleanly separates corrupted spectra before calibration.

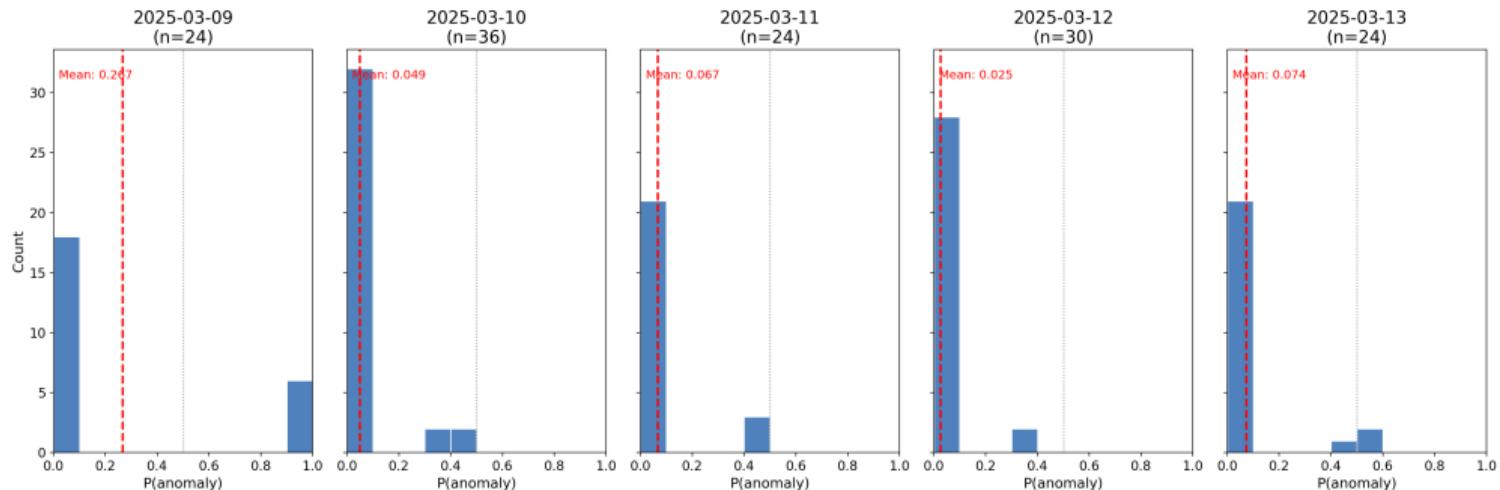


## Calibrator-level anomaly statistics

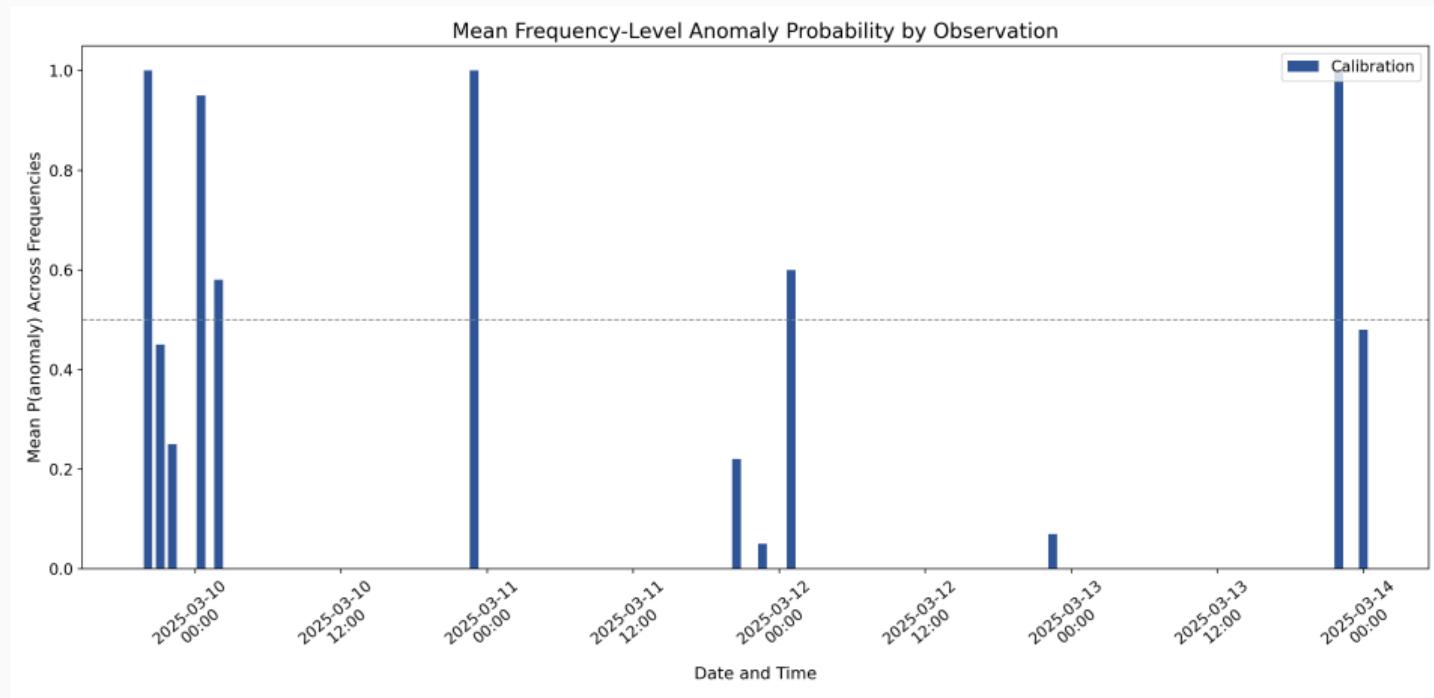


# Daily anomaly probability distributions

Distribution of Anomaly Probabilities by Date (Calibration Only)



# Observation-level anomaly tracking



# **End to end 21cm experiment simulation**

---

# Simulation pipeline for radiometer calibration

 **calibration-simulation** Private

[Watch](#) 1 [Fork](#) 0 [Star](#) 1

[main](#) [Branch](#) [Tags](#)  [Add file](#) [Code](#)

 **samleene** add more extensive codebase tests ✓ 50e82de · 8 minutes ago  78 Commits

 .github/workflows	add more extensive codebase tests	8 minutes ago
 calsim	add more extensive codebase tests	8 minutes ago
 data	fix dicke switching error	4 months ago
 debug_plots	add sin example	4 months ago
 examples	add more extensive codebase tests	8 minutes ago
 tests	add more extensive codebase tests	8 minutes ago
 .gitignore	cleanup	4 months ago
 CLAUDE.md	Add is_spectra feature for preserving frequency-dependent ...	2 hours ago
 LICENSE	modified: .gitignore	5 months ago
 MANIFEST.in	fix dicke switching error	4 months ago
 README.md	Add is_spectra feature for preserving frequency-dependent ...	2 hours ago
 pyproject.toml	fix dicke switching error	4 months ago
 requirements.txt	refactor	4 months ago
 setup.cfg	fix dicke switching error	4 months ago
 setup.py	fix dicke switching error	4 months ago

**About**

Calibration observation simulation codes for REACH

 [Readme](#)  
 [Apache-2.0 license](#)  
 [Activity](#)  
 [Custom properties](#)  
 [1 star](#)  
 [1 watching](#)  
 [0 forks](#)

**Releases**

No releases published [Create a new release](#)

**Packages**

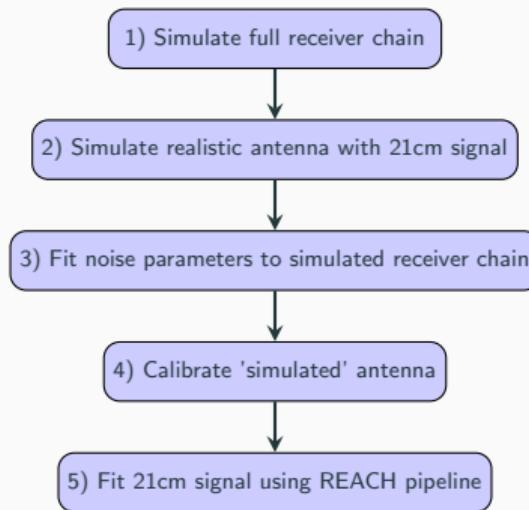
No packages published [Publish your first package](#)

**Languages**

 Python 100.0%

## End to end simulation

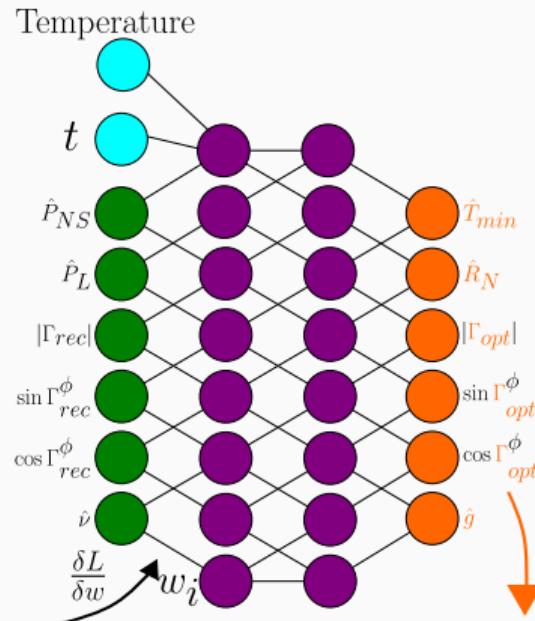
We have shown we can calibrate an internal source, we now test the method on as part of the broader system (simulated).



## **Modeling system drift from other complex system effects**

---

# Modeling environmental features



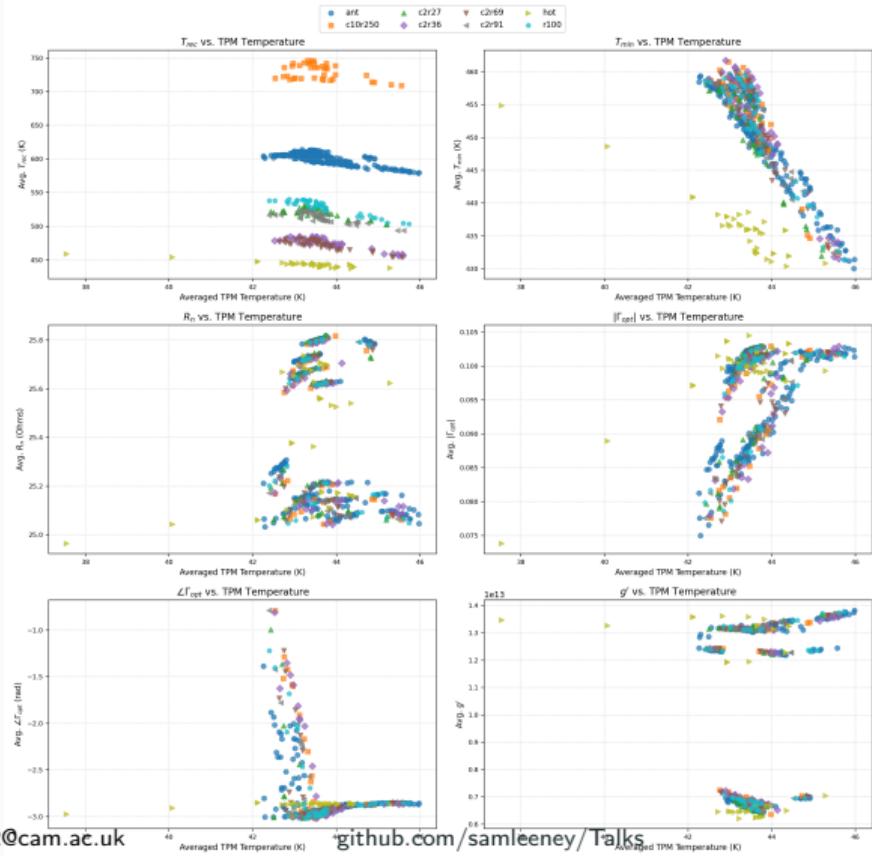
We can train on features that cannot be modeled analytically

$$\mathcal{L} = \frac{1}{n} \sum_i^n \frac{1}{\mathcal{P}_{src}^{out2}} \left[ \mathcal{P}_{src}^{out} - \mathcal{P}_{src}^{pred} \right]^2 \quad \leftarrow \quad \mathcal{P}_{src}^{pred} = gM \left( T_{src}^{in} + T_{min} + T_0 \frac{4R_N}{Z_0} \frac{|\Gamma_{src} - \Gamma_{opt}|^2}{(1 - |\Gamma_{src}|^2)(1 + |\Gamma_{opt}|^2)} \right)$$



# Fitting environmental features against fitted noise parameters

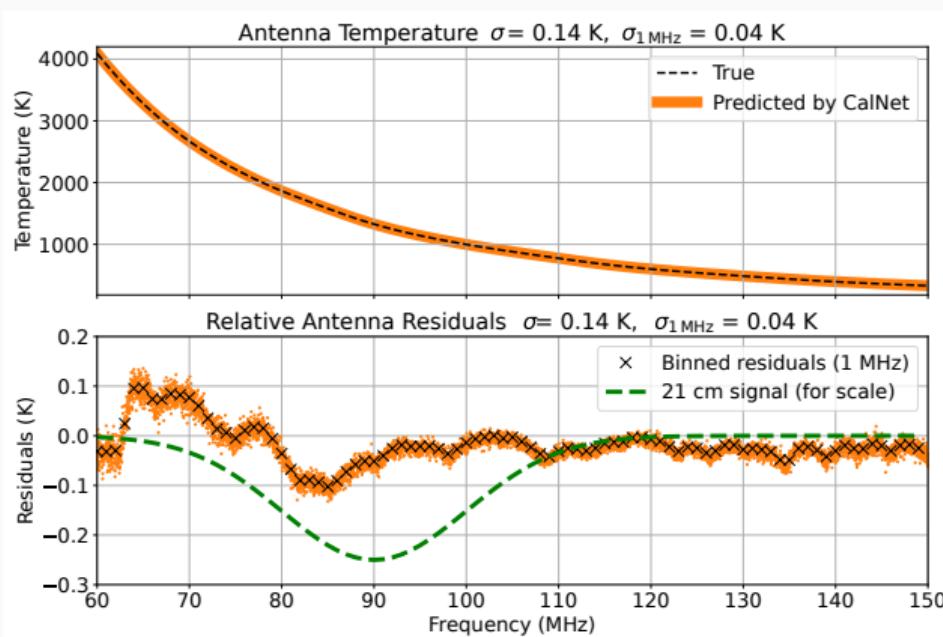
Correlate noise parameters with environmental data



## **End to end simulations**

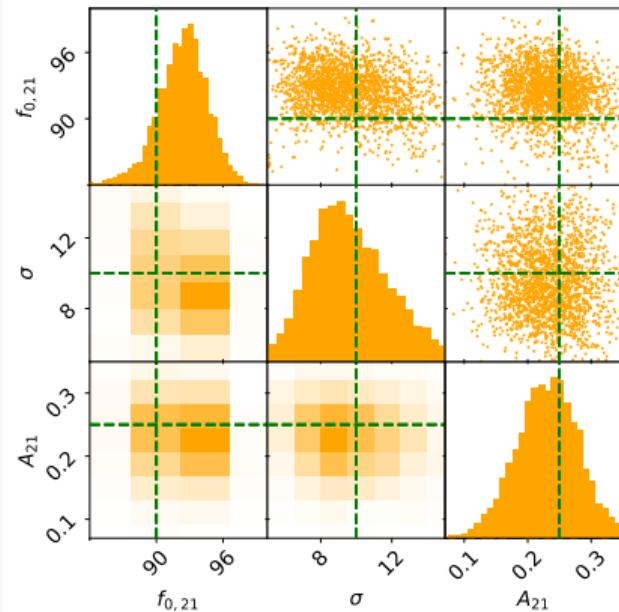
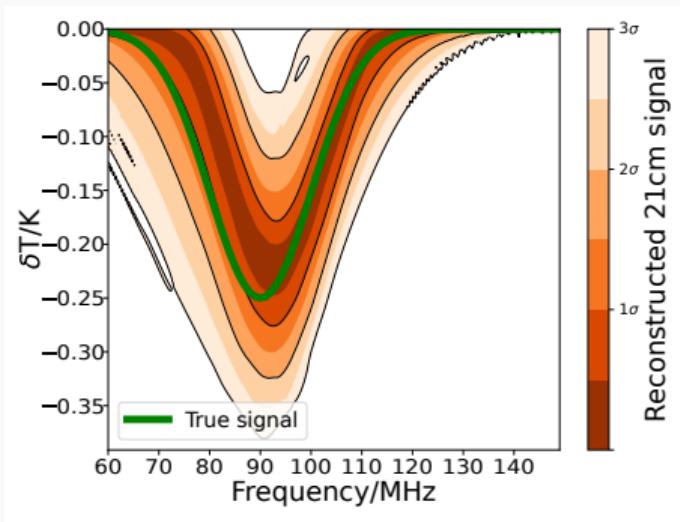
---

## Predicted vs True Antenna Temperature

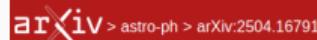


**Figure 5:** The top panel shows the predicted antenna temperature in orange, with the true temperature overlaid in black dashes

## Inferred 21cm Signal



# Read the paper...



Search...

Help | Adv

Astrophysics > Instrumentation and Methods for Astrophysics

[Submitted on 23 Apr 2025]

## Radiometer Calibration using Machine Learning

S. A. K. Leeney, H. T. J. Bevins, E. de Lera Acedo, W. J. Handley, C. Kirkham, R. S. Patel, J. Zhu, D. Molnar, J. Cumner, D. Anstey, K. Artuc, G. Bernardi, M. Bucher, S. Carey, J. Cavillot, R. Chiello, W. Croukamp, D. I. L. de Villiers, J. A. Ely, A. Fialkov, T. Gessey-Jones, G. Kulkarni, A. Magro, P. D. Meerburg, S. Mittal, J. H. N. Pattison, S. Pegwal, C. M. Pieterse, J. R. Pritchard, E. Puchwein, N. Razavi-Ghods, I. L. V. Roque, A. Saxena, K. H. Scheutwinkel, P. Scott, E. Shen, P. H. Sims, M. Spinelli

Radiometers are crucial instruments in radio astronomy, forming the primary component of nearly all radio telescopes. They measure the intensity of electromagnetic radiation, converting this radiation into electrical signals. A radiometer's primary components are an antenna and a Low Noise Amplifier (LNA), which is the core of the "receiver" chain. Instrumental effects introduced by the receiver are typically corrected or removed during calibration. However, impedance mismatches between the antenna and receiver can introduce unwanted signal reflections and distortions. Traditional calibration methods, such as Dicke switching, alternate the receiver input between the antenna and a well-characterised reference source to mitigate errors by comparison. Recent advances in Machine Learning (ML) offer promising alternatives. Neural networks, which are trained using known signal sources, provide a powerful means to model and calibrate complex systems where traditional analytical approaches struggle. These methods are especially relevant for detecting the faint sky-averaged 21-cm signal from atomic hydrogen at high redshifts. This is one of the main challenges in observational Cosmology today. Here, for the first time, we introduce and test a machine learning-based calibration framework capable of achieving the precision required for radiometric experiments aiming to detect the 21-cm line.

Comments: Under peer review for publication in Nature Scientific Reports as part of the Radio Astronomy collection

Subjects: Instrumentation and Methods for Astrophysics (astro-ph.IM); Cosmology and Nongalactic Astrophysics (astro-ph.CO); Artificial Intelligence (cs.AI)

Cite as: arXiv:2504.16791 [astro-ph.IM]

(or arXiv:2504.16791v1 [astro-ph.IM] for this version)

<https://doi.org/10.48550/arXiv.2504.16791> 

### Submission history

From: Samuel Alan Kossoff Leeney [[view email](#)]

[v1] Wed, 23 Apr 2025 15:10:25 UTC (25,900 KB)

Thank you!



SCAN ME

## Appendix

---

## Calibration Equation

Typically, substitute in the noise wave parameter equation here (gains cancel)

$$T_{\text{cal}}^* = T_{\text{NS}} \frac{P_{\text{cal}} - P_L}{P_{\text{NS}} - P_L} + T_L \quad (4)$$

Make some matching assumptions and re arrange:

$$\begin{aligned} T_s = & \color{red} T_{\text{NS}} \left( \frac{P_s - P_L}{P_{\text{NS}} - P_L} \right) \frac{|1 - \Gamma_s \Gamma_{\text{rec}}|^2}{1 - |\Gamma_s|^2} + \color{red} T_L \frac{|1 - \Gamma_s \Gamma_{\text{rec}}|^2}{1 - |\Gamma_s|^2} - \color{red} T_{\text{unc}} \frac{|\Gamma_s|^2}{1 - |\Gamma_s|^2} + \\ & - \color{red} T_{\text{cos}} \frac{\Re \left( \frac{\Gamma_s}{1 - \Gamma_s \Gamma_{\text{rec}}} \right) |1 - \Gamma_s \Gamma_{\text{rec}}|^2}{(1 - |\Gamma_s|^2) \sqrt{1 - |\Gamma_{\text{rec}}|^2}} - \color{red} T_{\text{sin}} \frac{\Im \left( \frac{\Gamma_s}{1 - \Gamma_s \Gamma_{\text{rec}}} \right) |1 - \Gamma_s \Gamma_{\text{rec}}|^2}{(1 - |\Gamma_s|^2) \sqrt{1 - |\Gamma_{\text{rec}}|^2}} \end{aligned} \quad (5)$$

Note: We end up with 5 parameters that need to be estimated to calibrate the system.

## Calculating the error

**By partial derivatives** To find the error in  $T_s$ , we propagate the errors in  $\Gamma_s$ ,  $\Gamma_{rec}$ ,  $P_L$ ,  $P_{NS}$ , and  $P_s$ :

$$(\Delta T_s)^2 = \left( \frac{\partial T_s}{\partial \Gamma_s} \Delta \Gamma_s \right)^2 + \left( \frac{\partial T_s}{\partial \Gamma_{rec}} \Delta \Gamma_{rec} \right)^2 + \left( \frac{\partial T_s}{\partial P_L} \Delta P_L \right)^2 + \left( \frac{\partial T_s}{\partial P_{NS}} \Delta P_{NS} \right)^2 + \left( \frac{\partial T_s}{\partial P_s} \Delta P_s \right)^2. \quad (10)$$

## Calculating the error

$$(\Delta T_s)^2 = \left( \frac{\partial T_s}{\partial \Gamma_s} \Delta \Gamma_s \right)^2 + \left( \frac{\partial T_s}{\partial \Gamma_{rec}} \Delta \Gamma_{rec} \right)^2 + \left( \frac{\partial T_s}{\partial P_L} \Delta P_L \right)^2 + \left( \frac{\partial T_s}{\partial P_{NS}} \Delta P_{NS} \right)^2 + \left( \frac{\partial T_s}{\partial P_s} \Delta P_s \right)^2. \quad (11)$$


---

$$\frac{\partial T_s}{\partial P_L} = T_{NS} \frac{|1 - \Gamma_s \Gamma_{rec}|^2}{1 - |\Gamma_s|^2} \cdot \frac{P_s - P_{NS}}{(P_{NS} - P_L)^2}, \quad (12)$$

$$\frac{\partial T_s}{\partial P_{NS}} = -T_{NS} \frac{|1 - \Gamma_s \Gamma_{rec}|^2}{1 - |\Gamma_s|^2} \cdot \frac{P_s - P_L}{(P_{NS} - P_L)^2}, \quad (13)$$

$$\frac{\partial T_s}{\partial P_s} = T_{NS} \left( \frac{1}{P_{NS} - P_L} \right) \frac{|1 - \Gamma_s \Gamma_{rec}|^2}{1 - |\Gamma_s|^2}. \quad (14)$$

$$\frac{\partial T_s}{\partial \Gamma_{rec}} = \frac{\partial A}{\partial \Gamma_{rec}} + \frac{\partial B}{\partial \Gamma_{rec}} + \frac{\partial D}{\partial \Gamma_{rec}} + \frac{\partial E}{\partial \Gamma_{rec}}. \quad (15)$$

$$\frac{\partial T_s}{\partial \Gamma_s} = \frac{\partial A}{\partial \Gamma_s} + \frac{\partial B}{\partial \Gamma_s} + \frac{\partial C}{\partial \Gamma_s} + \frac{\partial D}{\partial \Gamma_s} + \frac{\partial E}{\partial \Gamma_s}. \quad (16)$$

$$A = T_{NS} \left( \frac{P_s - P_L}{P_{NS} - P_L} \right) \frac{|1 - \Gamma_s \Gamma_{rec}|^2}{1 - |\Gamma_s|^2}, \quad (17)$$

$$B = T_L \frac{|1 - \Gamma_s \Gamma_{rec}|^2}{1 - |\Gamma_s|^2}, \quad (18)$$

$$C = -T_{unc} \frac{|\Gamma_s|^2}{1 - |\Gamma_s|^2}, \quad (19)$$

$$D = -T_{cos} \frac{\Re \left( \frac{\Gamma_s}{1 - \Gamma_s \Gamma_{rec}} \right) |1 - \Gamma_s \Gamma_{rec}|^2}{(1 - |\Gamma_s|^2) \sqrt{1 - |\Gamma_{rec}|^2}}, \quad (20)$$

$$E = -T_{sin} \frac{\Im \left( \frac{\Gamma_s}{1 - \Gamma_s \Gamma_{rec}} \right) |1 - \Gamma_s \Gamma_{rec}|^2}{(1 - |\Gamma_s|^2) \sqrt{1 - |\Gamma_{rec}|^2}}. \quad (21)$$

## Calculating the error

Using  $T_{NS} \frac{P_{cal}-P_L}{P_{NS}-P_L} + T_L$

$$(\Delta T_s)^2 = \left( \frac{\partial T_s}{\partial P_s} \Delta P_s \right)^2 + \left( \frac{\partial T_s}{\partial P_L} \Delta P_L \right)^2 + \left( \frac{\partial T_s}{\partial P_{NS}} \Delta P_{NS} \right)^2 \quad (22)$$

$$+ \left( \frac{\partial T_s}{\partial \Gamma_s} \Delta \Gamma_s \right)^2 + \left( \frac{\partial T_s}{\partial \Gamma_{rec}} \Delta \Gamma_{rec} \right)^2. \quad (23)$$

---

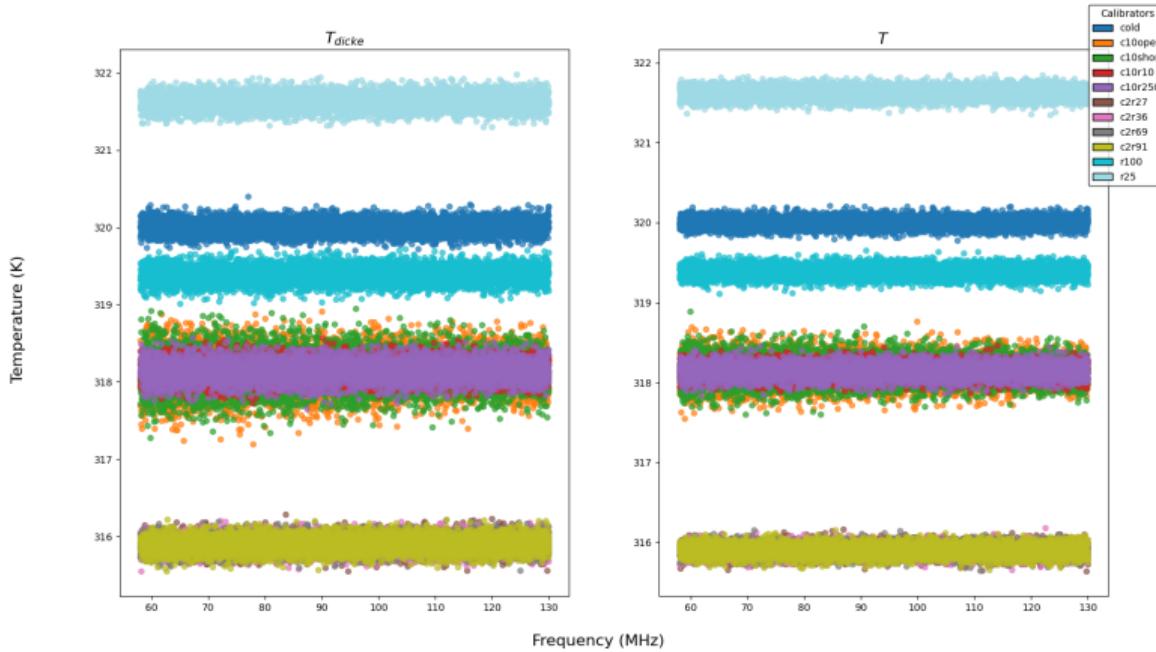
Using noise wave parameters only

$$(\Delta T_s)^2 = \left( \frac{\partial T_s}{\partial P_s} \Delta P_s \right)^2 \quad (24)$$

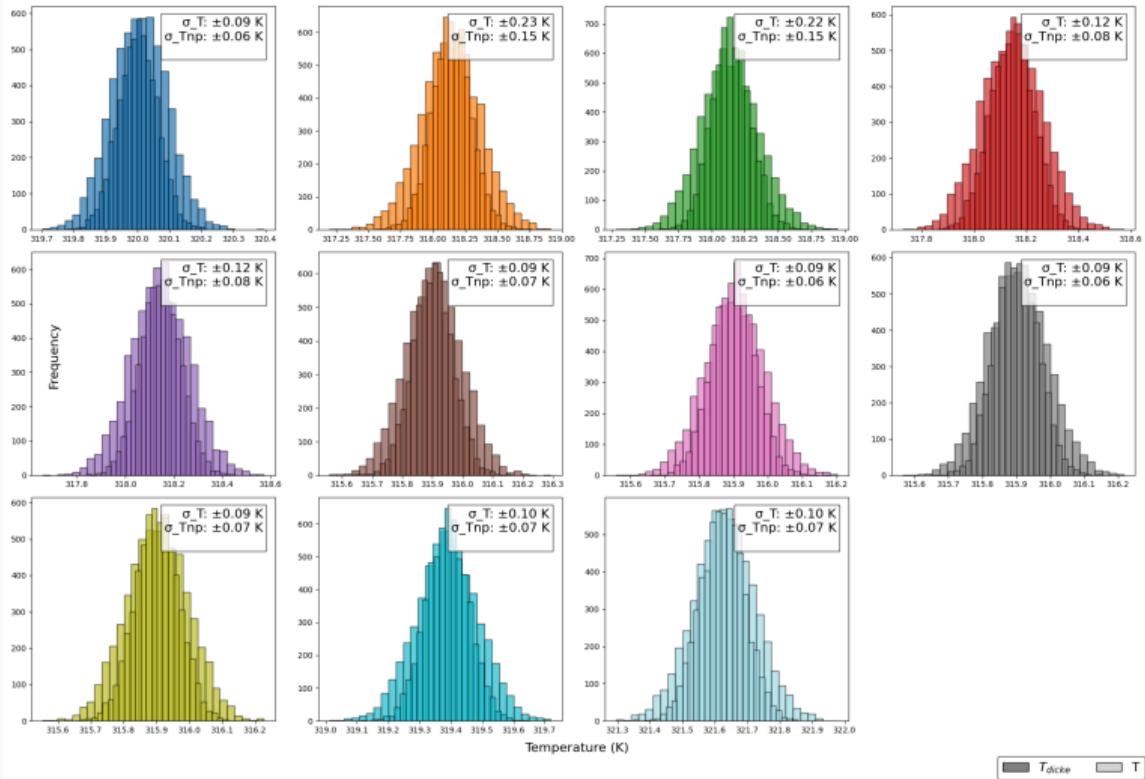
$$+ \left( \frac{\partial T_s}{\partial \Gamma_s} \Delta \Gamma_s \right)^2 + \left( \frac{\partial T_s}{\partial \Gamma_{rec}} \Delta \Gamma_{rec} \right)^2. \quad (25)$$

Note that this argument applies for both noise parameters and noise wave parameters.

There is more  
noise when using  
 $T_{\text{NS}} \frac{P_{\text{cal}} - P_L}{P_{\text{NS}} - P_L} + T_L$



Combined Histograms of  $T_{dicke}$  and T for Each Calibrator



Noise amplified  
by **30%** when  
using

$$T_{NS} \frac{P_{cal} - P_L}{P_{NS} - P_L} + T_L$$

Why not fit noise (wave) parameters directly?

Noise Parameter Equation:

$$P_{\text{out}}^{\text{src}} = \mathbf{g} M \left( T_{\text{in}}^{\text{src}} + \mathbf{T}_{\text{min}} + T_0 \frac{4R_N}{Z_0} \frac{|\Gamma_{\text{src}} - \Gamma_{\text{opt}}|^2}{(1 - |\Gamma_{\text{src}}|^2)(1 + |\Gamma_{\text{opt}}|^2)} \right) \quad (26)$$

---

Noise Wave Equation:

$$\begin{aligned} P_{\text{out}}^{\text{src}} = & \mathbf{g} \left[ \mathbf{T}_0 + \mathbf{T}_{\text{unc}} |\Gamma_s|^2 \left| \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{1 - \Gamma_s \Gamma_{\text{rec}}} \right|^2 \right. \\ & + T_s (1 - |\Gamma_s|^2) \left| \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{1 - \Gamma_s \Gamma_{\text{rec}}} \right|^2 + \mathbf{T}_{\cos} \Re \left( \Gamma_s \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{1 - \Gamma_s \Gamma_{\text{rec}}} \right) \\ & \left. + \mathbf{T}_{\sin} \Im \left( \Gamma_s \frac{\sqrt{1 - |\Gamma_{\text{rec}}|^2}}{1 - \Gamma_s \Gamma_{\text{rec}}} \right) \right] \end{aligned} \quad (27)$$

We still end up with 5 unknowns, as before.